

1.

```
C:\Users\seonh\Documents\Backup\Coding\ProgrammingC\DataStructu
re\241113>a
1 2 3 4 5
2
C:\Users\seonh\Documents\Backup\Coding\ProgrammingC\DataStructu
re\241113>a
2 4 6 8 10 12
10
```

insert 함수의 작동에 맞게 modifiedSearch 함수를 작성하는 것이 가장 어려웠으나 k가 현재 node의 key와 상태를 비교해서 각 케이스 별로 나누어 하나씩 코드를 완성했다.

2.

```
C:\Users\seonh\Documents\Backup\Coding\ProgrammingC\DataStructu
re\241113>a
S E S 1 3 4 5 8
C:\Users\seonh\Documents\Backup\Coding\ProgrammingC\DataStructu
re\241113>a
E S E S E 6 8 10 12 14 16 18 20
```

원소를 삭제하는 함수인 delete함수가 가장 큰 문제였다.

코드를 작성하기 전 BST에서의 삭제가 어떻게 이루어지는지에 대한 이해를 먼저 하고, 삭제하고자 하는 원소의 자식이 둘 다 NULL인지 한쪽만 NULL인지 자식이 두 개 존재하는지에 대한 케이스를 나누었다. 만약 자식이 두 개라면 삭제하고자 하는 원소를 해당 원소의 왼쪽 서브트리에서 가장 큰 값으로 바꾸고, 자식의 위치를 알맞게 조정하는 방법을 사용하여 코드를 완성했다.